

R_ControlFunctions.R

nicholasgotelli

Thu Feb 4 15:19:34 2016

```
# Control Functions in R
# 04 Feburary 2016
# NJG
```

```
# The If statement if (cond) expression
```

```
TellWeather <- function(sky="blue") {
if (sky=="blue") cat("The weather is clear!")
}
```

```
# Compund IF statement if(cond) {expr1 expr2.....}
```

```
TellWeather2 <- function(sky="blue") {
if (sky=="blue") {
  Temp <- sample(70:90,1)
  cat("The weather is clear!","\n", "The temperature is", Temp,"degrees!", "\n")
}
}
```

```
# Compund IF statement and ELSE IF statement if(cond) expr1 else alt.expr
```

```
TellWeather3 <- function(sky="blue") {
if (sky=="blue") {
  Temp <- sample(70:90,1)
  cat("The weather is clear!","\n", "The temperature is", Temp,"degrees!", "\n") } else {
  Temp <- sample(30:60,1)
  cat("The weather is not clear!","\n", "The temperature is", Temp,"degrees!", "\n")
}
}
```

```
# Compund IF statement and ELSE IF ELSE statement
```

```
TellWeather4 <- function(sky="blue") {
if (sky=="blue") {
  Temp <- sample(70:90,1)
  cat("The weather is clear!","\n", "The temperature is", Temp,"degrees!", "\n") } else if(sky=="gray") {
  Temp <- sample(30:60,1)
  cat("The weather is not clear!","\n", "The temperature is", Temp,"degrees!", "\n")
} else {cat("The weather is uncertain! Open the window!")
}
}
```

```
# being clear is better than being fast
```

```
TellWeather5 <- function(sky="blue") {
```

```

if (sky=="blue") {
  Temp <- sample(70:90,1)
  cat("The weather is clear!","\n", "The temperature is", Temp,"degrees!", "\n") }

if(sky=="gray"){
  Temp <- sample(30:60,1)
  cat("The weather is not clear!","\n", "The temperature is", Temp,"degrees!", "\n")
}

if(!(sky %in% c("blue","gray"))){
  cat("The weather is uncertain! Open the window!")
}
}

```

```

#IFELSE for useful vectorized operations
# ifelse(test,yes,no)

```

```

z <- runif(10)
ifelse(z>0.5,3,0)

```

```
## [1] 3 0 3 3 3 3 0 3 3 0
```

```
z
```

```
## [1] 0.63397897 0.42480195 0.79764139 0.71075025 0.52768912 0.71043850
## [7] 0.19673079 0.94479249 0.54165096 0.03506861
```

```

# The For Loop
# for (var in seq) expr

# Summing the squares of numbers with a simple for loop

```

```

z <- seq(1,3)
x <- 0
for (i in z) {
  x <- x + i^2
}
x

```

```
## [1] 14
```

```

# more typical
z <- 3
x <- 0
for (i in 1:z) {
  x <- x + i^2
}
# much easier to do this with a vector
z <- 1:3
z <-sum(z^2)
z

```

```
## [1] 14
```

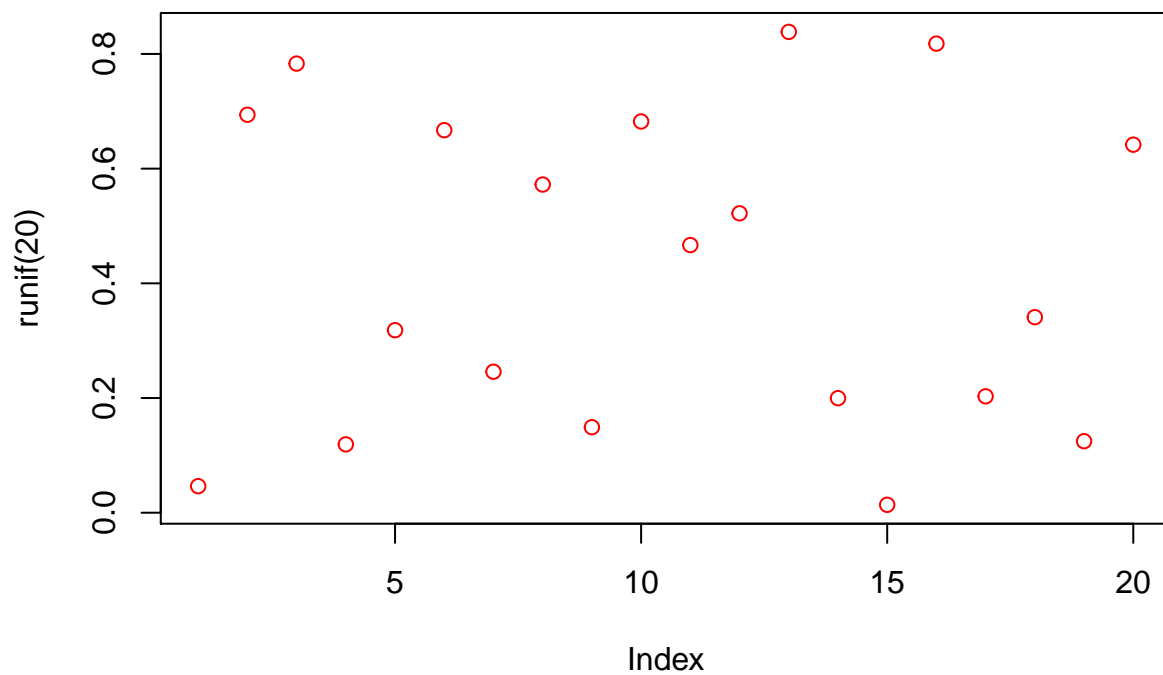
```
# sequence can be used to as an index of a vector
```

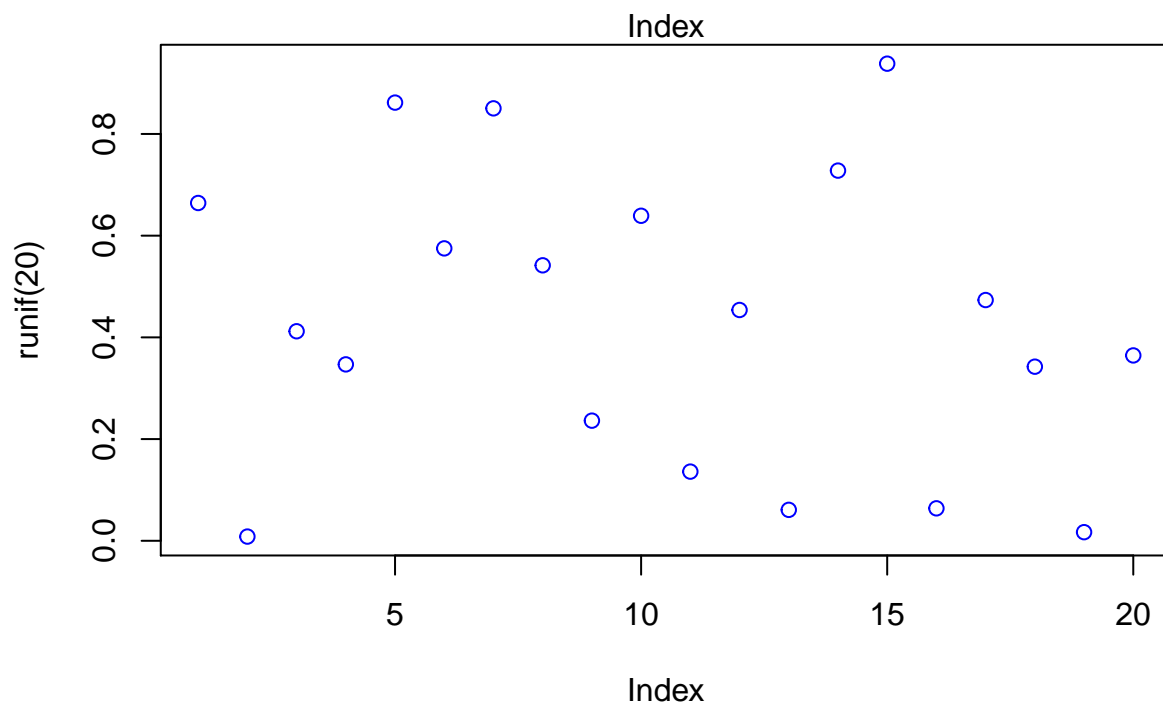
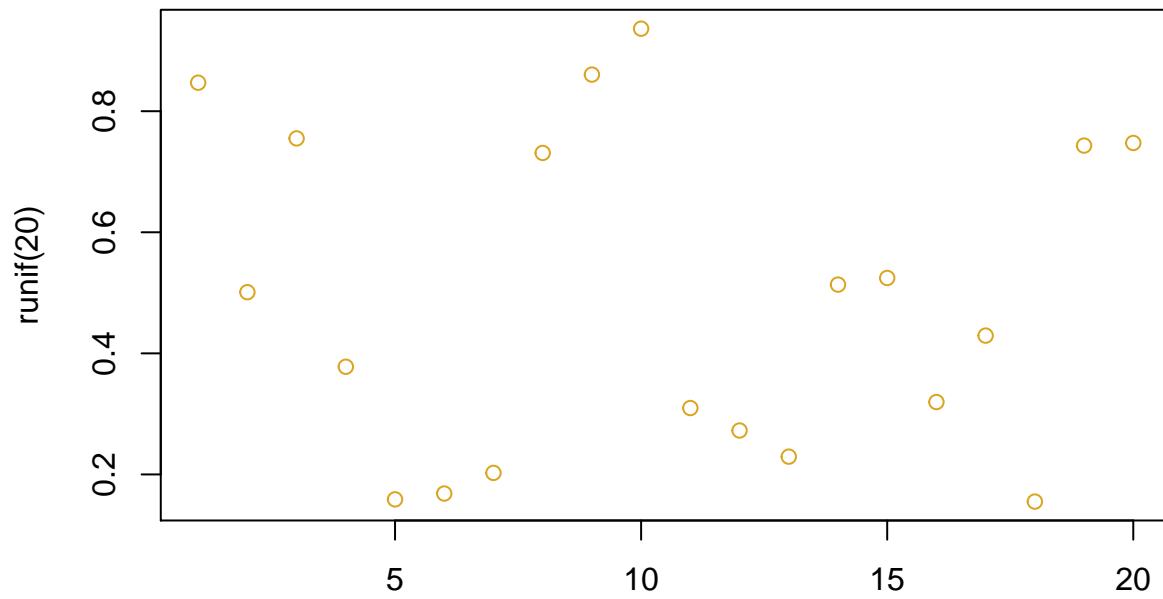
```
z <- c(3.3,4,10)
myvec <- vector(mode="numeric",length=length(z))
for (i in 1:length(z)) {
  myvec[i] <- z[i]^2
}
myvec
```

```
## [1] 10.89 16.00 100.00
```

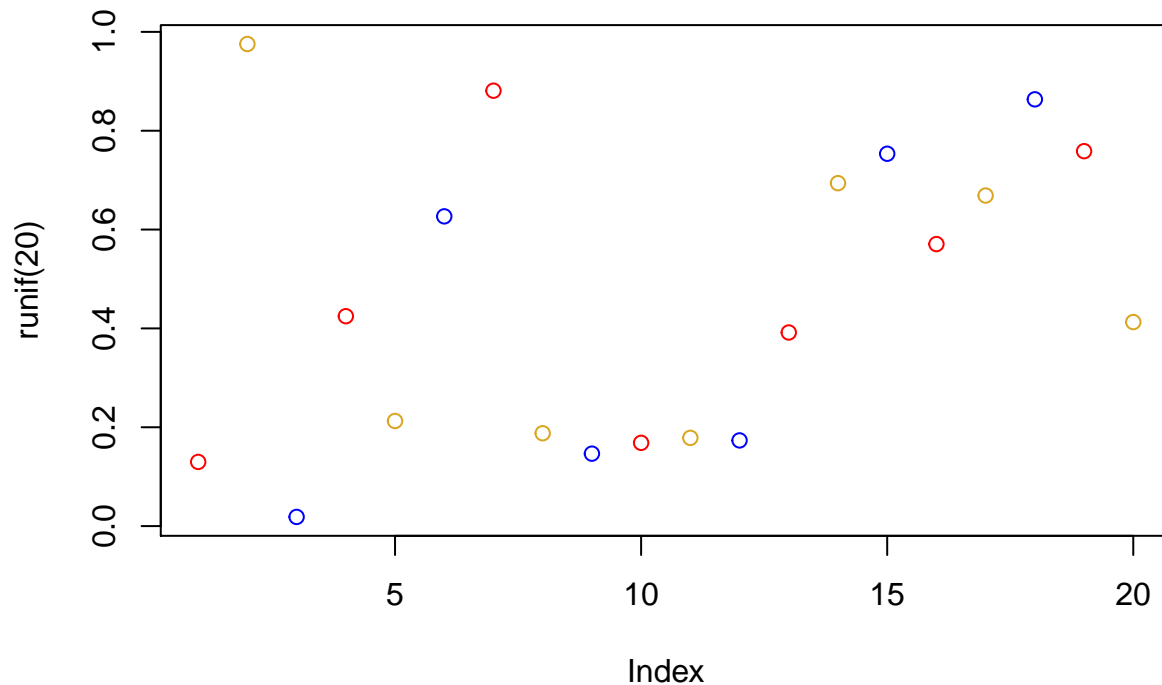
```
# text elements can also be cycled
```

```
z <- c("red","goldenrod","blue")
for (i in 1:length(z)){
  plot(runif(20),col=z[i])
}
```





```
# without a for loop, we only get one plot  
plot(runif(20),col=z)
```



double loops for matrix work

```
m <- matrix(1:12,nrow=4,byrow=TRUE)
for (i in 1:nrow(m)){
  for (j in 1:ncol(m)){
    z <- runif(1)
    m[i,j] <- m[i,j] + z
  }
}
```

results sensitive to placement of statements inside loops!
move z statement out of innermost loop

```
m <- matrix(1:12,nrow=4,byrow=TRUE)
for (i in 1:nrow(m)){
  z <- runif(1)
  for (j in 1:ncol(m)){
    m[i,j] <- m[i,j] + z
  }
}
```

move the z statement out of both loops

```
m <- matrix(1:12,nrow=4,byrow=TRUE)
z <- runif(1)
for (i in 1:nrow(m)){
  for (j in 1:ncol(m)){
    m[i,j] <- m[i,j] + z
  }
}
```

to add constant to columns, switch loops and operate on columns first

```
m <- matrix(1:12,nrow=4,byrow=TRUE)
for (j in 1:ncol(m)){
```

```
z <- runif(1)
for (i in 1:nrow(m)){
  m[i,j] <- m[i,j] + z
}
}
```